



# **The Dynamic Information Architecture System: A Simulation Framework to Provide Interoperability for Process Models**

**John R. Hummel and John H. Christiansen**

Advanced Computer Applications Center

Decision and Information Sciences Division

Argonne National Laboratory

Argonne, IL

Presented to:

Spring 2002 Simulation Interoperability Standards Organization

Orlando, FL

# Outline

## ➤ What is DIAS?

- Examples of DIAS Applications
- Summary

# What is DIAS?

- DIAS is an Object-based Software *Framework* for Modeling and Simulation Applications
- The DIAS Software Infrastructure Allows Many Disparate Simulation Models and Other Applications to Interoperate to Address a Complex Problem Based on the Context of the Specific Problem
- DIAS is Subject Domain Independent – the Integration of Models and Applications Gives it a Specific, Subject Domain “Flavor” (e.g. DEEM, IDLAMS, HealthSim, IOA, ...)

## What is DIAS? (Cont.)

- DIAS has been Awarded a US Patent (#6058387) and is Provided License-Free to US Govt. Agencies and Their Contractors
- DIAS has been Selected by the US Environmental Protection Agency as their Simulation Framework for Environmental Modeling Applications

# What are the Main Components of a DIAS Application?

- Software Objects That Represent the Real-World Entities in the Problem Space - Objects are Designed from a General Perspective in Order to Maximize Reuse
- Internal or External (i.e. Legacy) Simulation Models and Other Applications That Express the Dynamic Behaviors of the Domain Entities - DIAS Can Integrate Existing "Legacy" Models in Virtually Any Software Language
- Infrastructure Tools that Enable Context-Dependent:
  - Spatial Representation of Object Attributes (e.g. Point, 2-D, 3-D)
  - Multiple Representations of Object Behaviors

## What are the Main Components of a DIAS Application? (Cont.)

- API's have been Developed to Facilitate the Integration of Models for non-DIAS Experts
- The Unique DIAS Infrastructure Makes It Feasible to Build and Manipulate Complex Simulation Scenarios in Which Many Thousands of Objects Can Interact Via Dozens to Hundreds of Concurrent Dynamic Processes

# Q: Why is Simulation Context So Important?

## A: Entities Can be Simulated from Different Perspectives

• **"Blue" Warfighter's View:** A Weapons Platform to Attack From

• **Relevant Attributes:**

- Types and # of Weapons
- Speed
- Range
- etc.

• **"Red" Warfighter's View:** A Weapons Platform to Attack

• **Relevant Attributes:**

- Signature

• **Logistician's View:** An Item of Equipment to be Moved

• **Relevant Attributes:**

- Weight
- Size

• **Maintenance View:** An Item of Equipment to Repair and Service

• **Relevant Attributes:**

- POL Requirements
- Spare Parts Lists
- Repair Procedure Times
- ... etc.

• **Battlefield Commander's View:** A Platform to Direct Orders to

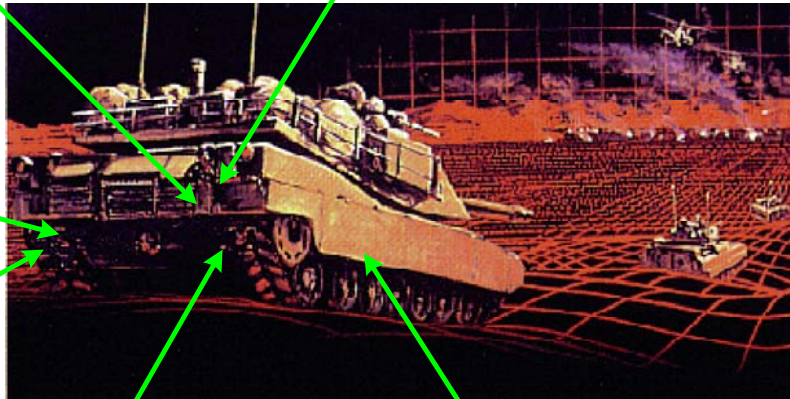
• **Relevant Attributes:**

- Position on the Battlefield
- Type of Platform
- Role to Play in a Course of Action (COA)
- ... etc.

• **Tank Commander's View:** A System to Direct Orders to

• **Relevant Attributes:**

- Position on the Battlefield
- State of Platform
- Current COA
- ... etc.



The Ultimate Goal is to Have a Single Object Design that can Adapt to these Different Perspectives

# DIAS Embraces and Extends the Object Paradigm

- Key Tenets of the “Object Paradigm” in Software Engineering are Embraced by DIAS:
  - Encapsulation: Objects Manage Their Own Attributes and Dynamic Behaviors, Thereby Promoting Clean, Modular Design
  - Inheritance: Object Subclasses Can "Inherit" Attributes and Behaviors From Parent Object Classes, Thereby Promoting Code **Reuse** and Extensibility

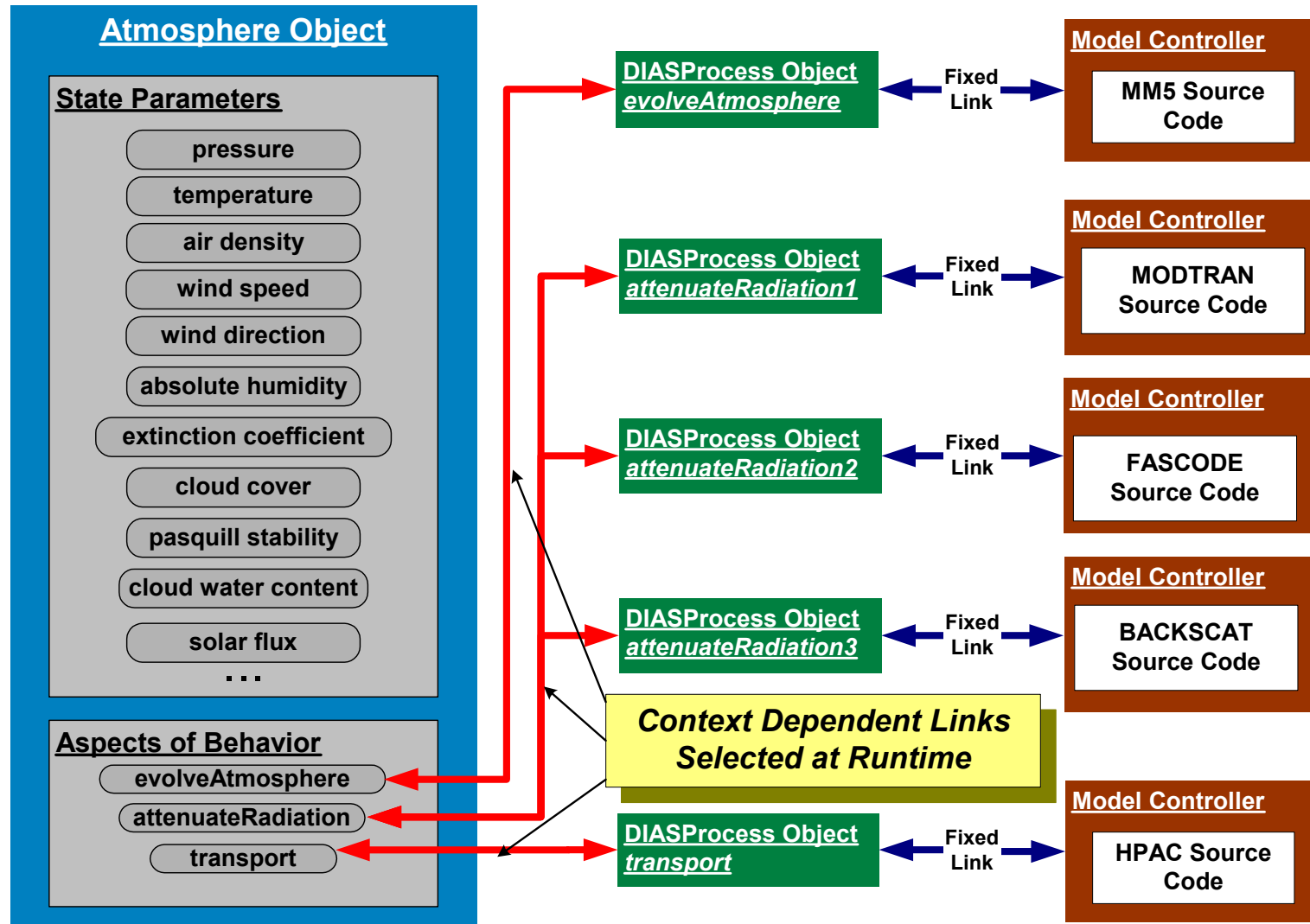


## DIAS Embraces and Extends the Object Paradigm (Cont.)

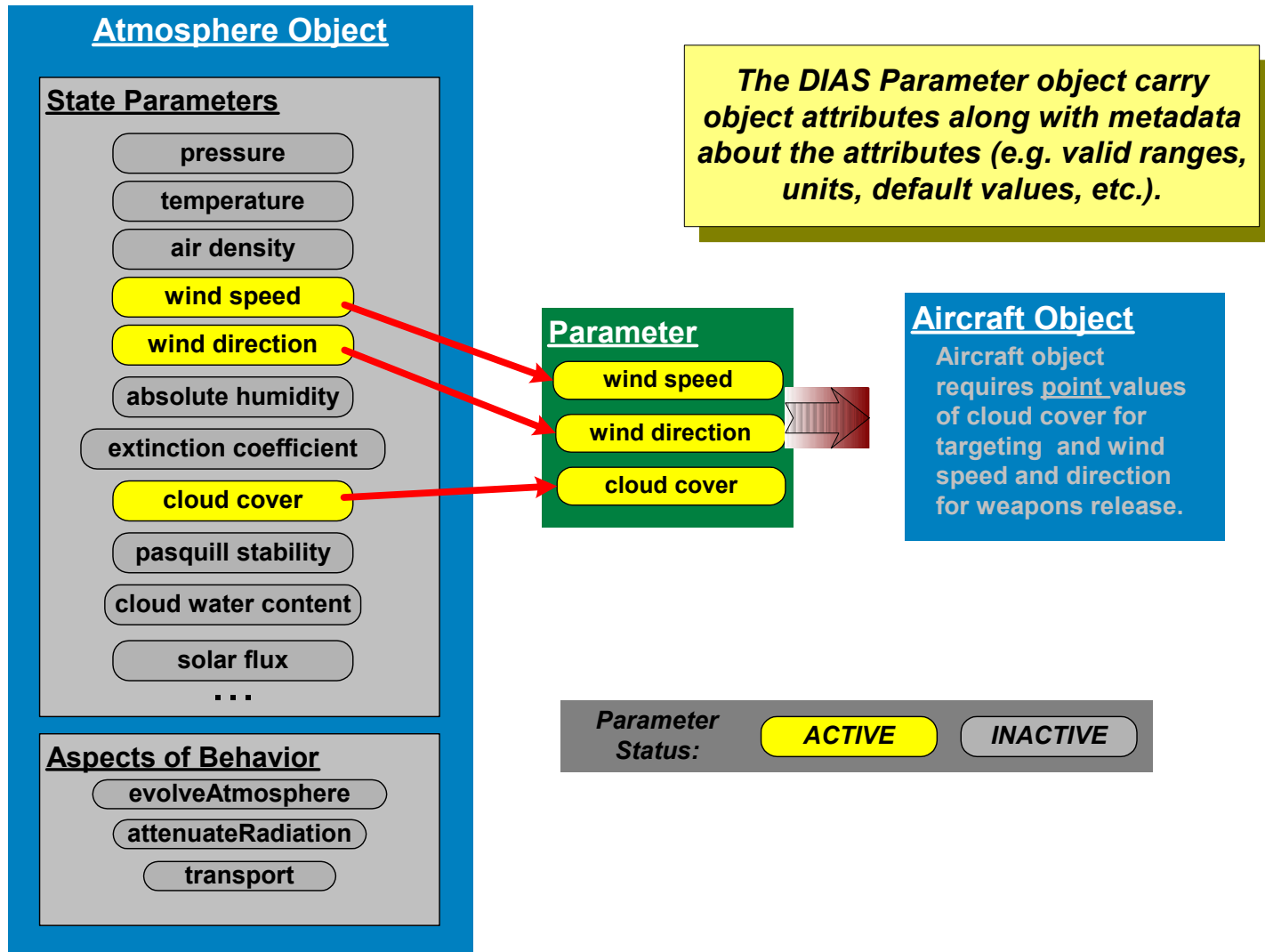
- DIAS Extends the Object Paradigm by *Abstraction* of the Objects' Dynamic Behaviors - the “*WHAT*” Is Separated From the “*HOW*”:
  - DIAS Object Definitions Contain an Abstract Description of the Behaviors (the *WHAT*) the Objects can Manifest, but No Details (the *HOW*) on how to Implement them
  - Separate Models / Applications Carry the Details of how to Implement Object Behaviors
  - Models Are Linked to Appropriate Domain Objects “on the Fly”, to Meet Specific Needs of a Given Simulation Context - Leading to Even Greater Flexibility and Extensibility



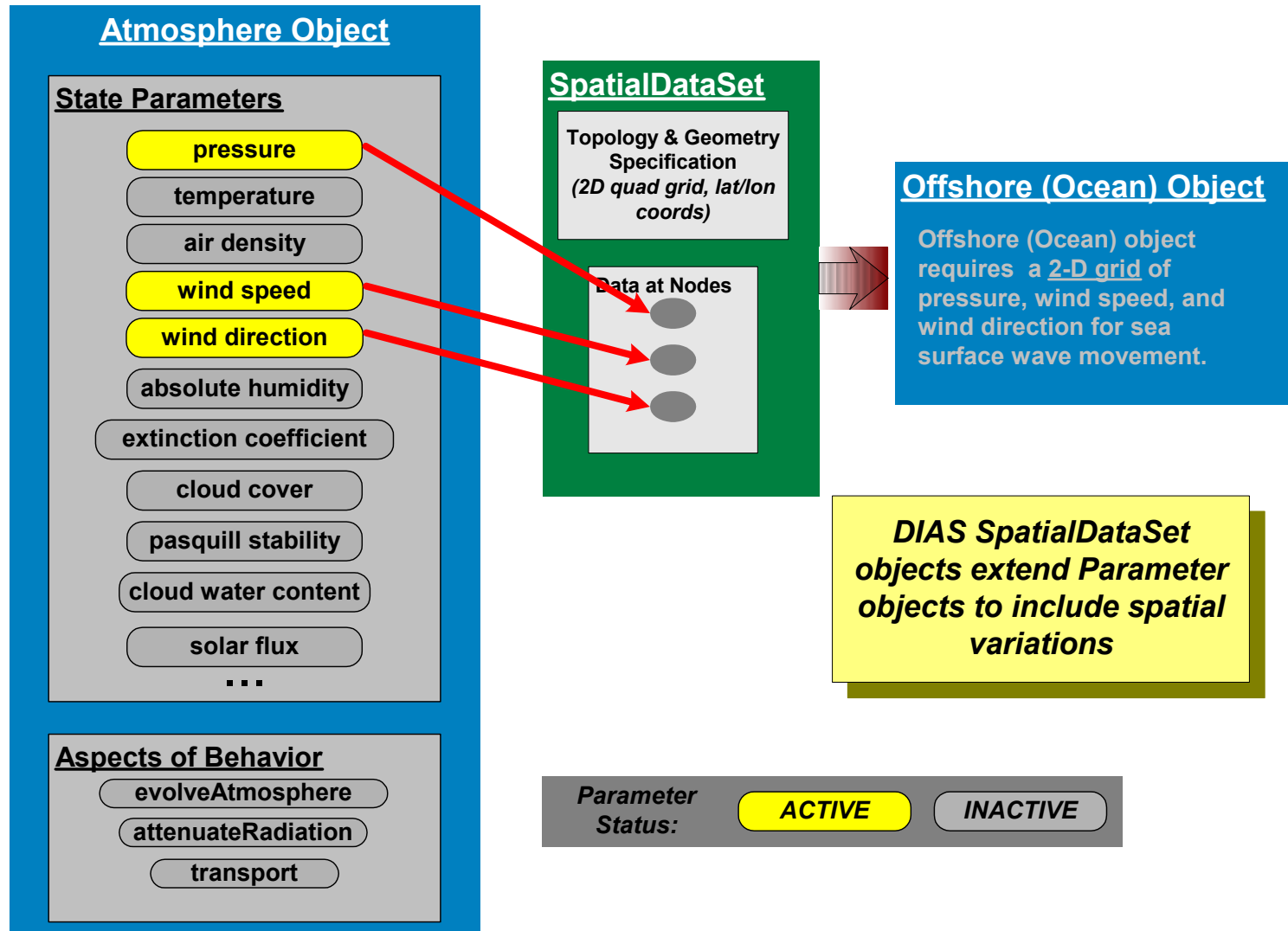
# DIAS Embraces and Extends the Object Paradigm: Context-Dependent Implementation of the *How*



# DIAS Embraces and Extends the Object Paradigm: Implementing Different Spatial Representations



# DIAS Embraces and Extends the Object Paradigm: Implementing Different Spatial Representations



# A Large (and Growing) DIAS Object Library Supports Construction of New Simulation Systems

## Object Classes in DIAS Object Library (Not an exhaustive list)

**Ephemeris**  
**Atmosphere**

**Hydrosphere**  
**GroundWater**  
**SurfaceWater**  
**HydroLayer**  
**WaterBody**  
**Stream**  
**ExternalStream**  
**Pool**  
**Offshore**  
**Nearshore**  
**SurfZone**  
**ExternalPool**  
**Shore**

**AreaOfInterest**  
**AvenueOfApproach**

**EarthSurface**

**Cover**  
**SoilCover**  
**WaterCover**  
**SnowIceCover**  
**SurfaceCover**  
**Developed**  
**Bare**  
**Wetland**  
**Grassland**  
**Forest**  
**Cane**  
**Plantation**  
**Cropland**  
**Vineyard**  
**Scrub**  
**Microterrain**

**TransportNet**  
**TransportLink**  
**TransportArc**  
**AirArc**  
**RailArc**  
**RoadArc**  
**TransportNexus**  
**Airfield**  
**Railyard**  
**HydroNexus**  
**CatchmentNexus**  
**OpenWaterNexus**  
**TransportNode**  
**TransportJunction**  
**AirJunction**  
**RailJunction**  
**RoadJunction**  
**HydroJunction**  
**TransportBoundary**  
**HydroBoundary**  
**Littoral**

**Artifact**  
**Structure**  
**Building**  
**Vehicle**  
**LandVehicle**

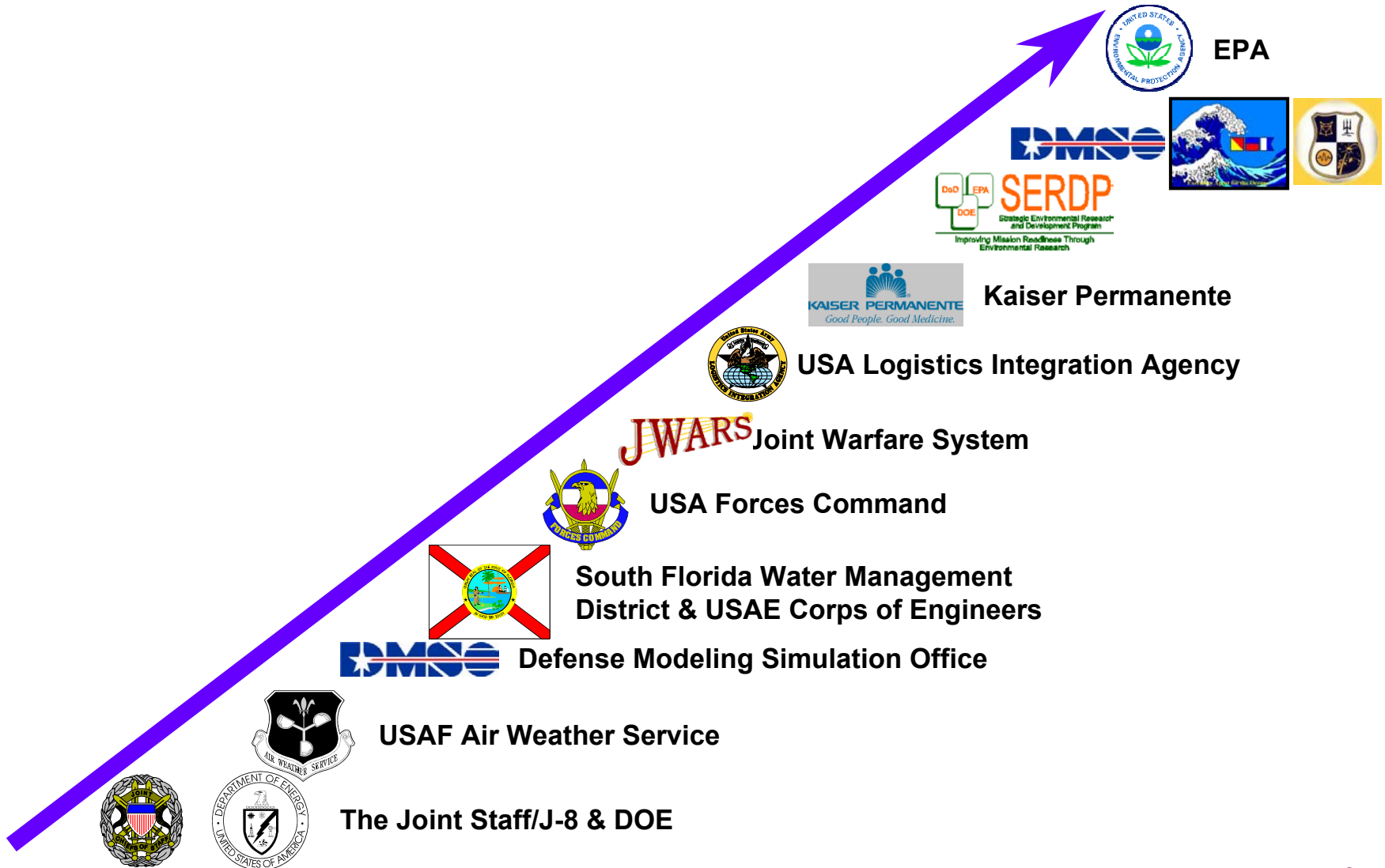
**ArcPathway**  
**Bridge, BridgeSpan**  
**Ferry**  
**Ford**  
**Tunnel**  
**NodeQualifier**  
**Constriction**  
**DropGateRail**  
**DropGateRoad**  
**HydroStructure**  
**HydroCulvert**  
**HydroGatedCulvert**  
**HydroGatedSpillway**  
**HydroPump**  
**HydroWell**  
**HydroDrain**  
**Levee**  
**LeveeSection**

**Person**  
**Physiology**  
**PhysiologicalSystem**  
**Organization**  
**MedicalDepartment**  
**Family**

# Outline

- What is DIAS?
- Examples of DIAS Applications
- Summary

# DIAS Evolution and Sponsor Applications



# DIAS Applications



**Dynamic weather and terrain influenced unit mobility, route planning and IPB analysis**



**Integrated oceanic systems simulation**



**Integrated land management and land use planning at military training bases**



**Health care: integrated physiological, clinical and logistical simulations**



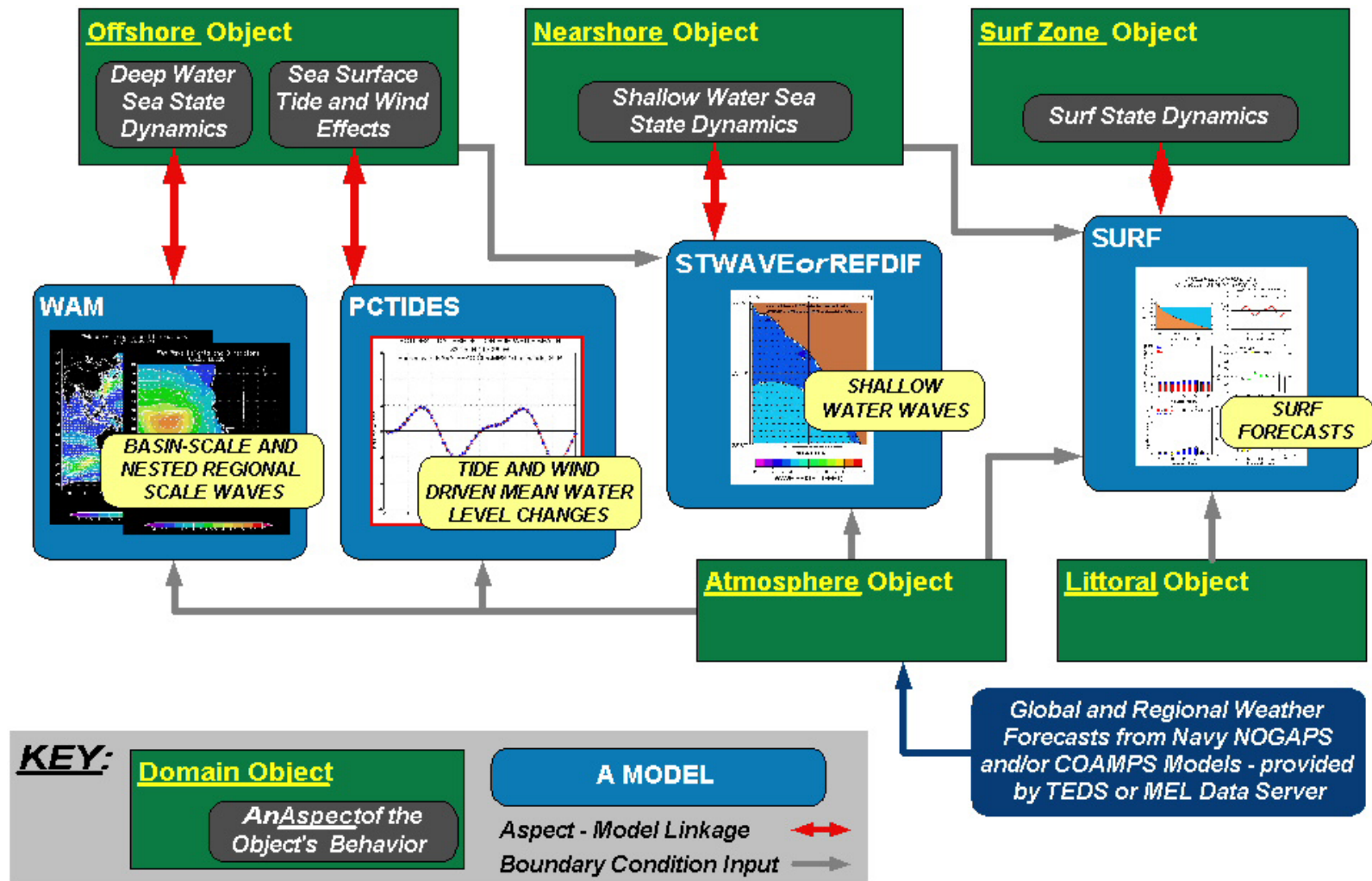
**Agricultural and social sustainability of ancient urban centers**



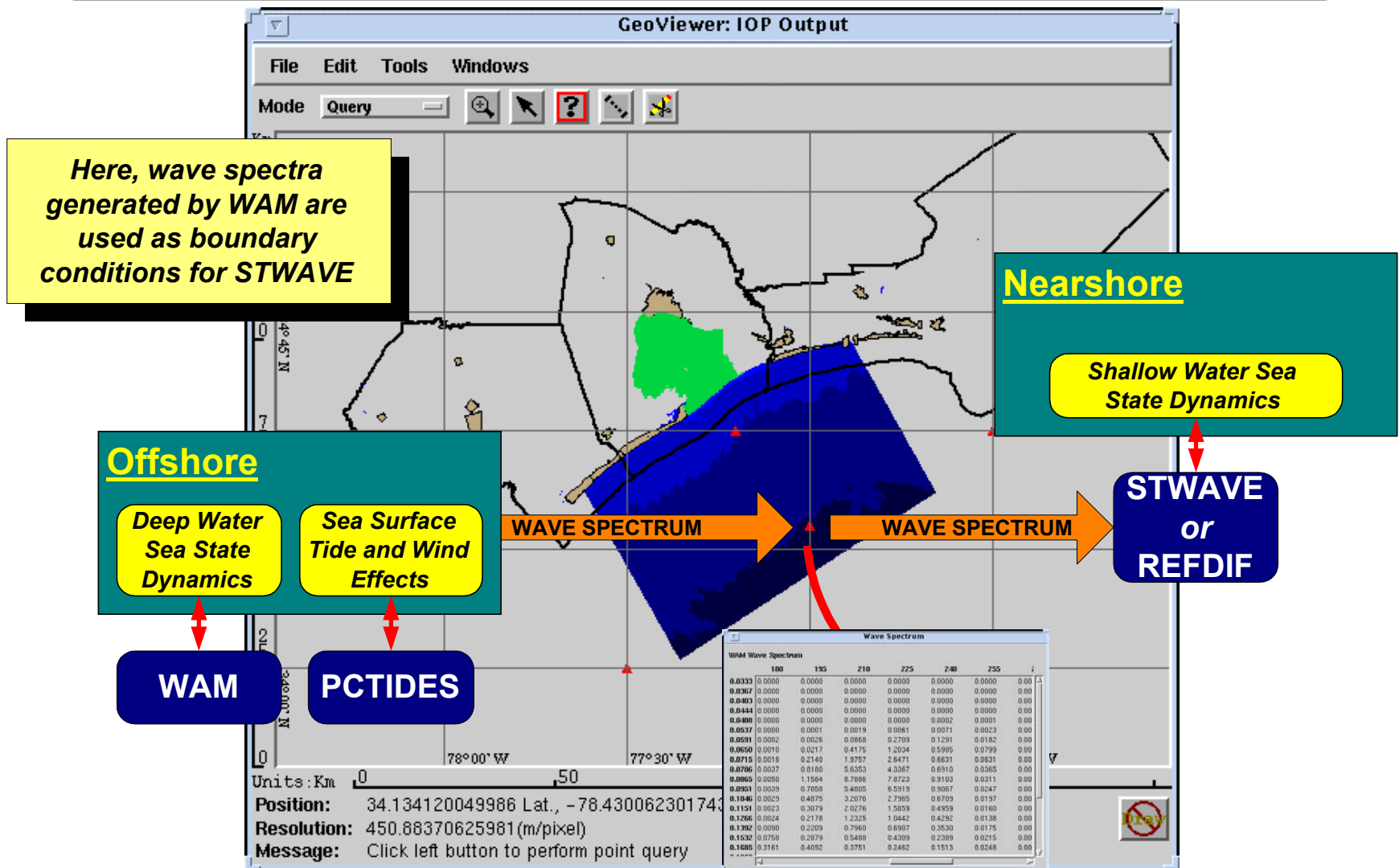
**Understanding the chemical "language" of cellular division**



# DIAS Applications: IOA – Integrated Ocean Architecture



# DIAS Applications: An IOA Application for Camp Le Jeune, NC



# Benefits of the Integrated Ocean Architecture

## **FASTER**

**Complex maritime simulations are far easier and quicker for analysts to set up**

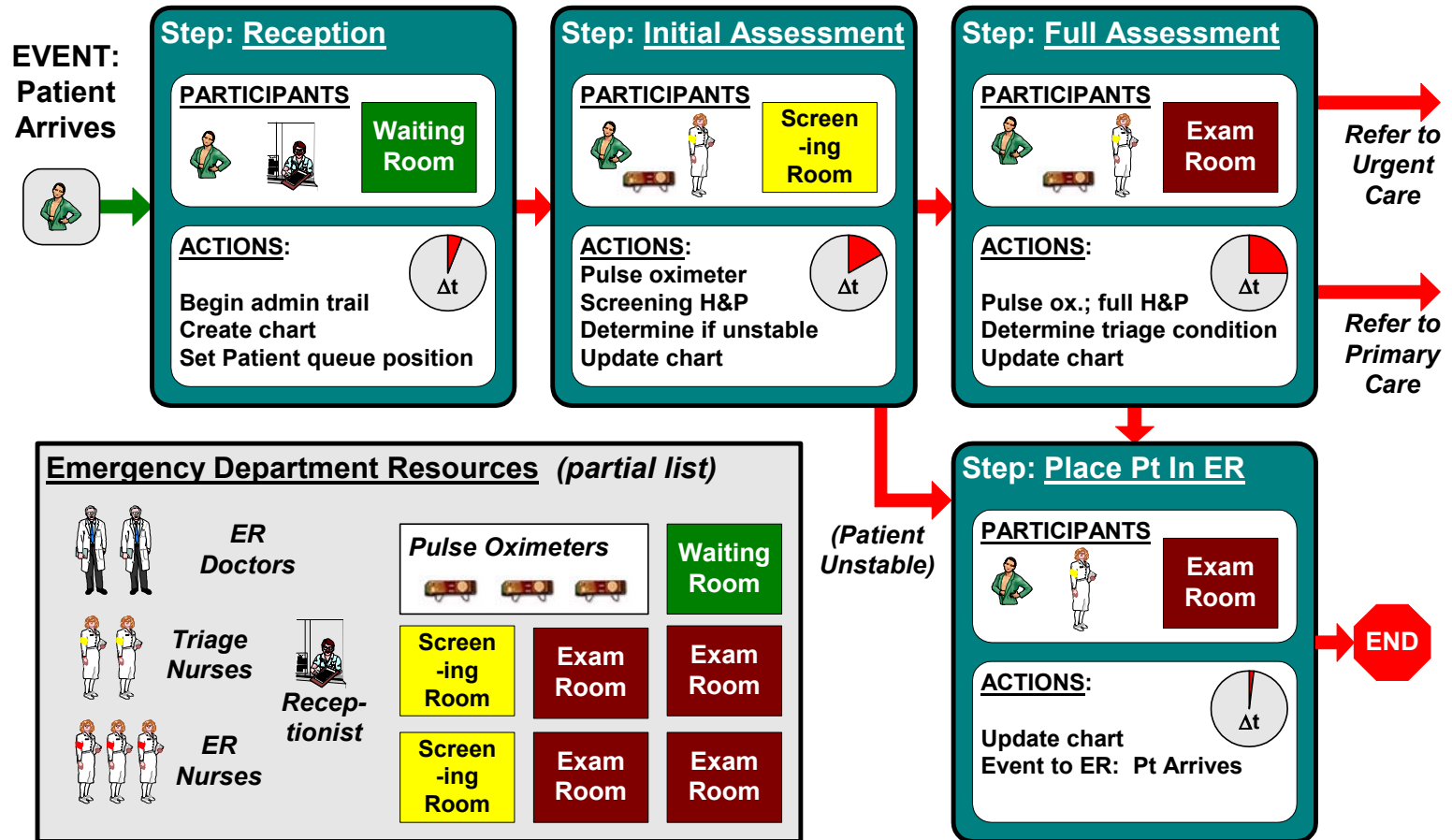
## **BETTER**

**DIAS / IOA common object repository ensures consistency of representation between models**

## **CHEAPER**

**The IOA virtual maritime environment can be modified and extended far more easily than most "model federations"**

# DIAS Applications: HealthSim – A Health Care Simulation System



# Outline

- What is DIAS?
- Examples of DIAS Applications

## ➤ Summary

## Summary

- DIAS is a Subject Domain Independent, Object-based Software *Framework* for Modeling and Simulation Applications
- DIAS Extends the Object Paradigm by *Abstraction* of the Objects' Dynamic Behaviors - the “*WHAT*” Is Separated From the “*HOW*”
- The Context of the Simulation Determines “*Who Does What to Whom and How*”

## For More Information

- DIAS Homepage: <http://www.dis.anl.gov/DIAS/>
- Call or e-mail:
  - John R. Hummel (630.252.7189), [jhummel@anl.gov](mailto:jhummel@anl.gov)
  - John H. Christiansen (630.252.3291), [jhc@anl.gov](mailto:jhc@anl.gov)